Issues Raised by Extreme Heterogeneity in Analytics

ASCR Extreme Heterogeneity Workshop

E. Wes Bethel, LBNL

23 Jan 2018

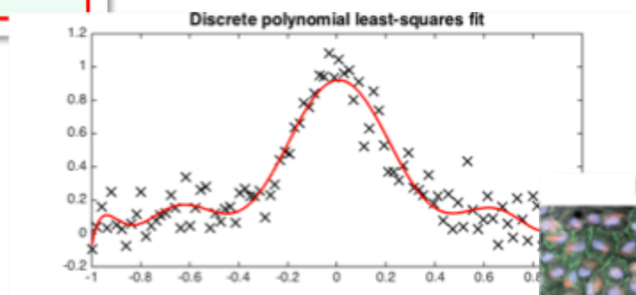
Data: Product or Source?

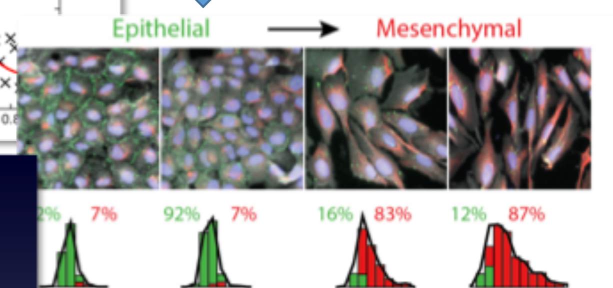
Data Analytics:
From data, derive a model, model parms, quantitative information

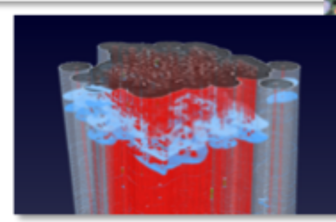
Modeling/simulation:
Solution to equations produces data.

Navier-Stokes momentum equation *(convective form)*

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla \bar{p} + \nu \nabla^2 \mathbf{u} + \tfrac{1}{3}\nu \nabla (\nabla \cdot \mathbf{u}) + \mathbf{g}.$$

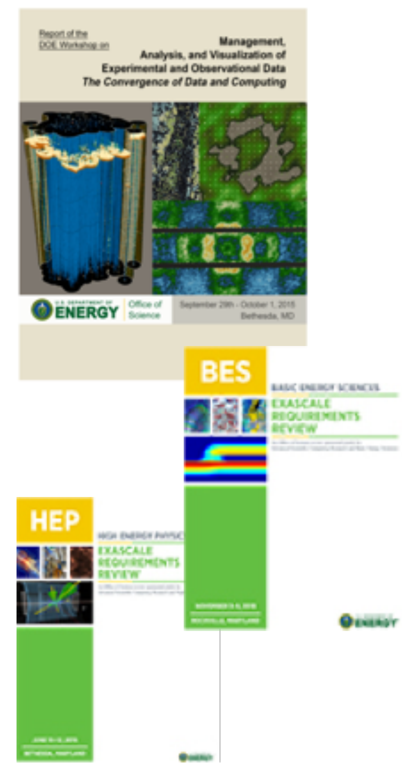






Heterogeneity in Use Cases, Data Sources

- Distributed collection of multi-modal sensors, produce curated data products (e.g., ARM/PNNL)
- Science user facility, individual experiments that produce data (e.g., ALS/LBNL, LCLS/SLAC, APS/ANL, SNS/ORNL, …)
    - Near-instrument processing
    - At-HPC center processing
    - Complex, multistage data-centric processing needs
    - Data lifecycle concerns
- Traditional computational science, simulation and modeling
    - Scale: Individual PI/project team, community-wide efforts
    - Data lifecycle concerns
- Lots of others:
    - Precision, personalized medicine
    - Cybersecurity, facilities operations



Heterogeneity in the Way Data is Used

- Datasets that are input to a method or aggregation
  - Hypothesis testing, discovery
- Collections that promote and facilitate scientific advances
  - Produced, shared by a community (e.g., AR, CMIP, SDSS, …)
- For training
  - Curated collections of labelled data for training supervised ML
- For optimization
  - Tune, optimize experiments
- For inference and prediction

- Note #1: the close symbiotic relationship (synergy) between data and compute
- Note #2: software and parameters are also "data"

Industry view (probably biased). More info: t.co/pXhCFOFvUz t.co/4OykMOLvNr. We need a similar diagram for science uses of data.

Heterogeneity in Methods and Software Environment (Partial View)

Analytics: Performance and Portability

- Individual methods:
  - Statistical/quantitative analysis, feature detection, learning, inference, visualization, …
  - Portable node-level parallelism, hybrid parallelism
  - Write once, run everywhere
    - X86, GPU, FPGA, TPU, NM, …
- Potential paths:
  - Traditional BSP design pattern:
    - MPI+X: where X provides for portable node-level parallelism
      - OpenMP 4.5: offload code onto accelerators (from FSD)
  - Alternate design pattern:
    - UDF in "hosted" environment or runtime system
      - Spark, TECA/DAGR, Legion, etc.
    - Traditional HPC vs. "Big Data" software stack

EH Trends

1. Increasing parallelism
2. Heterogeneous hardware acceleration
3. Data movement costs more than computation
4. Performance heterogeneity
5. New memory and storage technologies
6. User requirements

Analytics: Performance and Portability

- <u>Aggregations of methods</u>:
  - A sequence of individual methods
  - Data model and data movement issues
  - Resource marshaling and provisioning issues
  - Heterogeneous components:
    - OTS segmentation -> custom feature detection -> TensorFlow inference

- Potential paths:
  - Traditional workflow: Kepler, Tigris, etc.
  - Wide area (data movement): Globus, etc.
  - Analytics "environments":
    - TensorFlow [, Caffe, PyTorch, ...], Jupyter, ...
    - UDF-based (TECA/DAGR, ArrayUDF, Spark, ...)

- Note: these could be considered "workflow" issues, which Ewa will discuss next

EH Trends

1. Increasing parallelism
2. Heterogeneous hardware acceleration
3. Data movement costs more than computation
4. Performance heterogeneity
5. New memory and storage technologies
6. User requirements

Analytics: Reproducibility and Repeatability

- Desired outcome:
  - Yourself and others can reliably reproduce results of a computation
- What are the components?
  - Data, code, system environment (h/w, s/w)
    - Source code for methods: C++, Python, …
    - Environment: compiler, O/S, software environment (TensorFlow, PyTorch, MPI, VisIt, …)
    - DNN network topology, CART topology, etc.
    - Problem configuration: processing steps, ordering, parameters (layer weights, etc.), …
- Why is it important?
  - Integrity of scientific results
  - Basis for comparison of new methods: is the new method any better?
  - Preservation of knowledge
- <u>How are we going to do this?</u>

Closing Thoughts

- How to achieve performance and portability: 5, 10, 20 yrs?
  - Researcher/developer viewpoint
  - Scientist/consumer viewpoint
- Do we need abstractions for memory and storage hierarchy?
  - E.g., language-level constructs in CUDA
- Or do we let the language/compiler/environment take are of this?
  - PGAS memory model
  - Spark data/memory management
- Diversity in resources, policies and its impact on deployment, operations
- Tradeoffs between wanting to facilitate innovation, research and having a stable, predictable, maintainable ecosystem
- What can we "count on" being there for us in 5, 10, 20 yrs out?

EH Trends

1. Increasing parallelism
2. Heterogeneous hardware acceleration
3. Data movement costs more than computation
4. Performance heterogeneity
5. New memory and storage technologies
6. User requirements